Claims

1   1.   A multiple modulus conversion (MMC) method for obtaining a plurality
2   of index values associated with a plurality of moduli, for use in a communication system
3   having means configured to map frames of information bits onto predetermined
4   communication signal parameters, said method comprising:
5   obtaining an input;
6   representing the input as a plurality of sub-quotients;
7   performing a multiplication operation to multiply at least one of the sub-quotients
8   by a multiplicand; and
9   determining an index value associated with the multiplicand, the index value
10   being responsive to the inverse modulus multiplication operation.

1   2.   A method according to Claim 1, further comprising obtaining a
2   multiplicand that relates to the estimated inverse of a whole number, wherein performing
3   a multiplication operation includes multiplying at least one of the sub-quotients by the
4   multiplicand.

1   3.   A method according to Claim 2, wherein at least one pseudo remainder is
2   produced, the method further comprising checking at least one of the pseudo remainders
3   to determine whether the pseudo remainder falls within a predetermined range and, if it is
4   not within the predetermined range, adjusting the pseudo remainder such that the pseudo
5   remainder falls within the predetermined range.

1   4.   A method according to Claim 2, wherein at least one pseudo remainder is
2   produced, the method further comprising checking at least one of the pseudo remainders
3   to determine whether the pseudo remainder is larger than or equal to a predetermined
4   number and, if it is, subtracting the said number from the pseudo remainder.

1   5.   A method according to Claim 4, wherein the predetermined number is

2   subtracted from the pseudo remainder until the pseudo remainder is smaller than the

3   predetermined number.

1   6.   A method according to Claim 2, wherein at least one of the multiplicands

2   is pre-calculated, and stored in a table, the method further comprising obtaining at least

3   one multiplicand from the table.

1   7.   A method according to Claim 2, wherein the maximum number of digits

2   N_3 in the multiplicands, the maximum number of digits N_2 in the whole numbers, and the

3   maximum number of digits $n(j)$ in all but the most significant of the sub-quotients are

4   related by the inequality $N_3 > N_2 + n(j)$.

1   8.   A method according to Claim 2, wherein the multiplicand is calculated

2   according to the following formula: $C_i = floor(B^{N_3}/Y_i)$, where B is the base of the

3   numbering system, N_3 is the maximum number of digits in the multiplicands, and Y_i is

4   the whole number.

1   9.   A method according to Claim 2, wherein the input is represented as sub-

2   quotients in the form of $Q_{i-1} = Q_{i-1,0} + Q_{i-1,1}*B^{n(0)} + ... + Q_{i-1,k}*B^{n(0)+n(1)+..n(k-1)}$.

1   10.   A method according to Claim 8, wherein a sub-quotient of the output

2   pseudo-quotient is obtained by multiplying a portion of the input by the multiplicand C_i

3   according to the following formula: $Q_{i,j} = ((Q_{i-1,j}+R_{i,j+1}*B^{n(j)})*C_i)>>N_3$; and wherein the

4   output pseudo-remainder of the sub-quotient calculation is obtained according to the

5   following formula: $R_{i,j}=(Q_{i-1,j}+R_{i,j+1}*B^{n(j)})-(Q_{i,j}*Y_i)$, where B is the base number system.

1   11.   A method according to Claim 8, further comprising producing an output

2   remainder R_i and new quotient $Q_i = Q_{i,0} + Q_{i,1}*B^{n(0)} + ... + Q_{i,k}*B^{n(0)+n(1)+..n(k-1)}$.

1   12.   A multiple modulus conversion (MMC) encoder comprising:

2　an input configured to receive input data represented by a number of digits;

3　　　　　a multiplication means for multiplying two numbers; and

4　　　　　a processor configured to represent the input as a plurality of sub-quotients, to

5　multiply at least one sub-quotient by a multiplicand, and to produce an index value

6　associated with the multiplicand, the index value being responsive to the multiplication

7　operation.

1　　　　13.　　A modulus encoder according to Claim 12, wherein the multiplicand is

2　related to the estimated inverse of a whole number.

1　　　　14.　　A modulus encoder according to Claim 13, wherein the processor is

2　further configured to generate a pseudo remainder, and further configured to check

3　whether the pseudo remainder falls within a predetermined range.

1　　　　15.　　A modulus encoder according to Claim 13, wherein the processor is

2　further configured to generate a pseudo remainder, and further configured to check

3　whether the pseudo remainder is larger than or equal to a predetermined number, and

4　further configured to subtract the predetermined number from the pseudo remainder if the

5　pseudo remainder is larger than the predetermined number.

1　　　　16.　　A modulus encoder according to Claim 15, wherein the processor is

2　configured to keep subtracting the predetermined number from the pseudo remainder

3　until the pseudo remainder is smaller than the predetermined number.

1　　　　17.　　A modulus encoder according to Claim 13, further including storage

2　means to store at least one multiplicand.

1　　　　18.　　A modulus encoder according to Claim 13, wherein the processor is

2　configured to handle multiplicands with a maximum number of digits N_3, and is further

3　configured to handle moduli with a maximum number of digits N_2, and is further

4　configured to handle inputs represented as sub-quotients with a maximum number of

5   digits $n(j)$ in all but the most significant of the sub-quotients, wherein these quantities are

6   related by the inequality $N_3 > N_2 + n(j)$.

1   19.   A modulus encoder according to Claim 13, further including a inverse

2   estimate calculation means, wherein the inverse estimate is calculated according to the

3   following formula: $C_i = floor(B^{N_3}/Y_i)$, where B is the base of the number system, N_3 is

4   the maximum number of digits in the multiplicands, and Y_i is a whole number.

1   20.   A modulus encoder according to Claim 13, wherein the input is configured

2   to obtain an input value in the form $Q_{i-1} = Q_{i-1,0} + Q_{i-1,1}*B^{n(0)} + ... + Q_{i-1,k}*B^{n(0)+n(1)+..n(k-1)}$.

1   21.   A modulus encoder according to Claim 18, wherein the processor is

2   configured to obtain a sub-quotient of an output pseudo-quotient by multiplying a portion

3   of the input by an estimated inverse modulus according to the following formula: $Q_{i,j} =$

4    $((Q_{i-1,j}+R_{i,j+1}*B^{n(j)})*C_i)>>N_3$; and wherein the processor is further configured to obtain a

5   pseudo remainder according to the following formula: $R_{i,j}=(Q_{i-1,j}+R_{i,j+1}*B^{n(j)})-(Q_{i,j}*Y_i)$,

6   where B is the base of the numbering system.

1   22.   A modulus encoder according to Claim 18, wherein the processor is

2   further configured to generate an output remainder R_i and new quotient $Q_i = Q_{i,0} +$

3    $Q_{i,1}*B^{n(0)} + ... + Q_{i,k}*B^{n(0)+n(1)+..n(k-1)}$.